Pergamon

Computers Math. Applic. Vol. 36, No. 4, pp. 21–36, 1998

Printed in Great Britain
0898-1221/98 $19.00 + 0.00

PII: S0898-1221(98)00138-2

Interior Improvement of Piecewise Linear Interpolants Defined Over Delaunay Triangulations

D. LASSER
Center of Applied Mathematics
Technical University of Darmstadt
Hochschulstrassel, D-64289 Darmstadt, Germany

T. STÜTTGEN
Surveying and GIS Group
Ziegler Informatics, Duisburg, Germany

(Received and accepted December 1997)

Abstract—Given a set of points and corresponding function values we construct a piecewise linear C^0 interpolant defined over the Delaunay triangulation of the data set. The approximation quality of the interpolant is then improved by detecting *badly shaped* interior triangles, adding suitable points and updating the triangulation to form the new Delaunay triangulation and the new interpolant. Some of the developed schemes and repeated application of the idea can yield very impressive improvements of mean, maximum, and root mean square errors, as well as, of contour line plots. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords—Delaunay triangulation, Piecewise linear interpolation, Interior improvement.

1. INTRODUCTION

The scalar-valued $2D$ *Scattered Data Interpolation problem* (briefly SDI problem), many applications have to deal with, states as follows: given N distinct and noncollinear data (abscissae) $\mathbf{x}_i = (x_i, y_i) \in \mathbf{R}^2$, $i = 1(1)N$, and associated ordinates (e.g., function values or measured values) F_i, $i = 1(1)N$, find a function $z = f(\mathbf{x})$ such that $f(\mathbf{x}_i) = F_i$, $i = 1(1)N$. The SDI problem has been addressed by many authors. Some relevant survey articles on the subject are [1–3] and [4, Section 9]. Possible approaches to solve the SDI problem are, first, Shepard's inverse distance method and variations of it, second, radial basis function methods such as Hardy's multiquadric method, Duchon's thin plate splines, and Franke's thin plate splines in tension, and third, the triangle-based so-called FEM methods. The later are subject of this paper.

FEM methods work via a two-step procedure. First, a triangulation of the convex hull of the $2D$ data set $\{\mathbf{x}_i\}_{i=1(1)N}$ is constructed such that the vertices $\mathbf{P}_i$ of the triangulation coincide with the $\mathbf{x}_i$. Then for each triangle a function-valued surface patch is defined which interpolates given data (ordinates and possibly also derivatives) at vertices $\mathbf{P}_i$. In the simplest case, we get a piecewise linear C^0 continuous surface which interpolates only function values at vertices. For

This research is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) by Grant LA 665/6-1 and by Grant LA 665/6-3. The two authors would like to thank the German Research Foundation for its support.

higher order smoothness between neighbouring patches C^r (or GC^r) continuity conditions have to be enforced. This requires either the specification of a certain number of derivatives at the data or the estimation of them in a preprocessing step.

Quality, measured in terms of visual appearance, smoothness, accuracy, etc., of the interpolant depends on the triangle surface scheme used, the continuity order of patches, the accuracy of estimated derivative data, and on the triangulation. It is well known that derivatives (and triangulation, of course) have more influence on the global interpolant than the specific local surface scheme (i.e., a polynomial degree for example) or the order of continuity of patches (i.e., C^1 or C^2 continuity for example, see [5,6]).

While there are several papers investigating various methods to estimate derivatives, see [4] for a survey, not too much is done so far to improve the triangulation. For given data many different triangulations can be built up and the result, i.e., the triangulation created, often depends on the order in which data are processed. Therefore, it is common to base FEM interpolants on Delaunay triangulations, T^D. One reason for doing this is that the Delaunay triangulation is a globally optimal triangulation: several criteria (edge swapping criteria such as the max-min-angle and the circle criteria) exist for constructing the Delaunay triangulation and, the result is independent on the order in which data are processed. A second reason is that Delaunay triangulations create triangles as equiangular as possible and avoid long and thin triangles. Such triangles are regarded as bad for interpolation because error bounds for interpolation on triangles increase as the triangles become long and thin [7–9].

In the present paper, which is mainly based on [10], we are investigating possible improvements by altering the triangulation. To simplify calculations, we are restricting ourselves to interpolation from the space of piecewise linear functions defined on the Delaunay triangulation, $f_{T^D,F}$. A linear function is uniquely defined by its values at the three vertices of a triangle and therefore, no derivative estimation is required for the construction of this interpolant. The only degree of freedom in the determination of the PLI (*Piecewise Linear Interpolant*) is in the choice of the triangulation. But for reasons given above we decide to use the Delaunay triangulation of the data set.

Modification of the interpolant can be done by adding appropriately calculated points and corresponding function values to the data set, then update the triangulation to result again in a new, augmented data modified Delaunay triangulation, T^A, and form the new PLI based on T^A. Now, the questions are: how do we decide to add points, at which positions do we add points, and how do we calculate the new points and their corresponding function values?

Since some long and thin triangles near the boundary of the convex hull of the triangulation cannot be optimized, it is obvious that the approximation near the boundary is in many cases very poor relative to the approximation away from the boundary. Therefore, in [11] we described criteria and methods to detect and modify bad boundary triangles of the Delaunay triangulation to build up with a PLI. While some of the tested methods did not work too well, others were very successful and resulted in impressive improvements of mean, maximum and RMS (*Root Mean Square*) errors. To define these error measurements the PLI was used as an approximation to an underlying test surface, $F(x,y)$, and $3D$ data $(\mathbf{x}_i, F_i)$ were regarded as being sampled from that test surface, i.e., $F_i = F(x_i, y_i)$. The goal was to obtain by the interpolation $f(x,y)$ a good approximation to that test surface.

While boundary improvement is clearly of primarily interest, the approximation property of a PLI might also be improved by modification of interior bad triangles of the Delaunay triangulation. A triangle is defined as an interior triangle if it has three neighbors, i.e., if non of the triangle edges is part of the convex hull of the data set.

In this paper, we propose, test, and compare several methods, to detect bad interior triangles of the Delaunay triangulation and to generate additional data (abscissae and function values). They are supposed to reflect preliminary information about the underlying function. Numerical results and contour plots presented in Section 3 illustrate the success of some of these schemes in

improving the approximation quality of the PLI. Especially one of the suggested modifications via multiquadrics seems to be very suitable for improving the approximation in the mean, the maximum and the RMS errors. Repeated application of the algorithm using different methods can result in further improvements.

2. INTERIOR CORRECTION

The data-modified Delaunay triangulation is generated starting from an initial Delaunay triangulation, T^D, of the original input data. The initial Delaunay triangulation originates from a preprocessing step. It is created iteratively according to Lawson's algorithm [12]: first, we perform presorting through Euclidean distance, construct an initial triangle, and then add one point at a time, always maintaining a locally optimal triangulation. The later is insured by a LOP (*Local Optimization Procedure*) also suggested by Lawson, [12]: the *circumscribed circle criterion*. Lawson's LOP swaps diagonals of convex quadrilaterals in the triangulation until the globally optimal Delaunay triangulation has been created. The LOP converges after a finite number of edge swaps has been performed. Presorting and organization of the algorithm guarantees that, as a by-product, the convex hull of the data set is produced. The algorithm is capable of inserting additional points at any position and updating the triangulation using Lawson's LOP.

First, we need criteria to detect and to classify triangles as bad interior triangles and, second, we need methods to correct these triangles. We will present and test three detection criteria and five modification schemes, some of which result in very impressive improvements.

Starting from the initial Delaunay triangulation our algorithm traverses the triangle list using one of the three detection criteria described below in Section 2.1 to identify so-called bad interior triangles. If an interior triangle has been marked as bad, one additional point is calculated eventually and then added to the data set through one of the five different data adding schemes described in Section 2.2. After the Delaunay triangulation has been updated the new triangle list is traversed until there are no more bad interior triangles found with respect to the initially chosen detection criterion.

Throughout this paper we use the following notation: let $\mathbf{P} = (\mathbf{P}, z^P)$ denote a $3D$ point (a space point) while $\mathbf{P} = (x, y)$ denotes a $2D$ point (a domain point). Especially let $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ specify the vertices of an interior triangle under examination and let $\mathbf{S}$ be the barycenter of triangle $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$. Note, since $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is an interior triangle, vertices $\mathbf{Q}_i$ of the three direct neighbor triangles are always existent (though two of them might coincide), while some of the six neighbor points $\mathbf{R}_i$ might be not available in the data set (see Figure 1). Analogously, let **A**, **B**, **C**, **S**, etc., refer to the corresponding $3D$ data points.

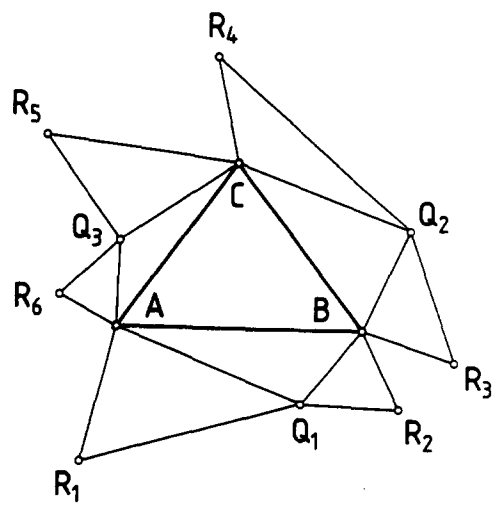


Figure 1. Triangle $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$, its direct neighbors $\mathbf{Q}_i$ and vertices $\mathbf{R}_j$ of their neighbors.

2.1. Detection of Badly Shaped Interior Triangles

To avoid that too many additional data are added to the triangulation and especially to avoid numerical problems a SAT *(Surface Area Test)* is performed first, before a detection criterion is applied to an interior triangle. The SAT tests if an interior triangle is that small that adding an extra point probably yields numerical problems but will not improve the approximation (therefore, the SAT is not needed in case of the surface area criterion described below which pinpoints very large triangles). Tests with randomly created data sets showed that the triangle area A has to fulfill $A > 0.002$ to result in a numerically stable algorithm. Interior triangles who passed the SAT finally enter the detection-modification-pipeline. Here, first, one of the three following detection criteria can be picked to work with to identify badly shaped interior triangles. Each criterion can be controlled via a constant threshold variable T. The criteria are as follows.

- **SA (Surface Area) criterion.** Let $\bar{A}$ be the mean surface area of triangles of the Delaunay triangulation. An interior triangle is classified as a bad triangle if

$$A > \bar{A}T_A, \quad \text{with } T_A > 1, \tag{1}$$

  with triangle surface area A and user defined constant area threshold T_A. The SA criterion is motivated by the fact that a relatively large domain triangle (indicated by $T_A = 3$ for example) might reflect missing input information and therefore, the PLI might give a poor approximation of the underlying function.
- **TA (Turning Angle) criterion.** Let ϕ be the turning angle of two adjacent triangles calculated via the scalar product of the upward oriented unit normal vectors of the two triangle planes. An interior triangle T with adjacent triangles T_i, T_j, and T_k is classified as a bad triangle if one of the three angles ϕ_i, ϕ_j, or ϕ_k between corresponding triangle normals, i.e., $\phi_i = \cos^{-1}(\mathbf{N} \cdot \mathbf{N}_i)$, etc., is larger than a user defined constant angle threshold, T_ϕ. Thus, if (note, it is $0 \leq \phi < \pi/2$)

$$\min\{\cos\phi_i, \cos\phi_j, \cos\phi_k\} < \cos T_\phi. \tag{2}$$

  The TA criterion (cf. the ABN criterion of [13]) is motivated by the fact that the underlying surface is supposed to be smooth relative to the data, i.e., $F(x, y)$ does not oscillate too much between data points. This means that the turning angles of normals of adjacent triangles of the PLI should not be too big.
- **STA (Stochastic TA) criterion.** For an automatic determination of a possible T_ϕ value in dependence of the given data, first, we determine for each of the interior edges of the triangulation angle ϕ_i, calculate the mean value $\bar{\phi}$, and the variation σ_ϕ. Angle threshold T_ϕ is then defined to be

$$T_\phi = \bar{\phi} + k\sigma_\phi \tag{3}$$

  with a suitably chosen positive constant k. We decided for the value

$$k = 1 + \frac{\sigma_\phi}{\bar{\phi}}. \tag{4}$$

  This is motivated as follows: first, according to the definition of σ_ϕ as the mean deviation of the mean value $\bar{\phi}$, it should be $k > 1$, otherwise too many triangles would be marked.

  Second, if σ_ϕ is of large value, there must be a few angles which are much larger than $\bar{\phi}$. In that case k should be large too to be able to detect these large values. On the other side, if σ_ϕ is of small value, k should be small too, otherwise no modifications would be done at all. Therefore, it should be $k \sim \sigma_\phi$.

  Third, if $\bar{\phi}$ is large, angles ϕ_i are large too, and k rather should be small in that case since for a large k value no modifications might be done at all. On the other side, if $\bar{\phi}$ is small then angles ϕ_i are small too, and k rather should be large in that case since for a small k value too many modifications might be done which actually might result not in improvement but the opposite effect. Therefore, it should be $k \sim 1/\bar{\phi}$.

Note, while the SA criterion usually addresses interior triangles further away from the convex hull, TA and STA criteria typically locate interior triangles near the convex hull. Obviously, for increasing threshold values fewer bad triangles are found.

2.2. Computation of Additional Data Points

The schemes described next add points in the interior of domain triangles and estimate the corresponding function values. It turned out that some of the schemes do not perform very well. On the other hand, others result in drastic improvements of the quality of the approximation. This will be demonstrated by the numerical experiments presented in Section 3.

- **ZEB-1 (z-value extrapolation barycenter) scheme.** Barycenter $\mathbf{S} = (\mathbf{S}, z^S)$ is added to the data set where z^S is found as a weighted sum of extrapolations along lines of the neighbouring triangles by the following procedure.
  1. Determine the barycenter $\mathbf{S}$ of triangle $\triangle(\mathbf{A}, \mathbf{B}, \mathbf{C})$.
  2. Define $\mathbf{K}_1$ as the intersection point between the line defined by $\mathbf{Q}_1$ and $\mathbf{S}$ and the line defined by $\mathbf{A}$ and $\mathbf{B}$. Calculate the z coordinate of $\mathbf{K}_1$ through linear interpolation or extrapolation along triangle edge **AB**:

$$z^{K_1} = z^A + t\left(z^B - z^A\right), \tag{5}$$

where

$$t = \begin{cases} \dfrac{x^{K_1} - x^A}{x^B - x^A}, & x^B - x^A \neq 0, \\ \dfrac{y^{K_1} - y^A}{y^B - y^A}, & \text{otherwise.} \end{cases}$$

  3. Analogously for auxiliary points $\mathbf{K}_2$ and $\mathbf{K}_3$ (see Figure 2).
  4. Calculate z value z_1^S through linear extrapolation along the line defined by $\mathbf{Q}_1$ and $\mathbf{K}_1$:

$$z_1^S = z^{Q_1} + t\left(z^{K_1} - z^{Q_1}\right), \tag{6}$$

where

$$t = \begin{cases} \dfrac{x^S - x^{Q_1}}{x^{K_1} - x^{Q_1}}, & x^{K_1} - x^{Q_1} \neq 0, \\ \dfrac{y^S - y^{Q_1}}{y^{K_1} - y^{Q_1}}, & \text{otherwise.} \end{cases}$$

  5. Analogously for z_2^S and z_3^S.
  6. Define z value z^S as a weighted sum of z values z_i^S $(i = 1, 2, 3)$, where the weighting w_i is a Shepard-like inverse distance weighting involving the distance between $\mathbf{Q}_i$ and $\mathbf{S}$, i.e., in domain space:

$$z^S = \sum_{i=1}^{3} w_i \, z_i^S, \tag{7}$$

where

$$w_i = \frac{1/(\|\mathbf{Q}_i - \mathbf{S}\|)}{\sum_{j=1}^{3}(1/\|\mathbf{Q}_j - \mathbf{S}\|)}.$$

- **ZEB-2 scheme.** Auxiliary points $\mathbf{K}_i = (\mathbf{K}_i, z^{K_i})$ are defined and z-values z_i^S of barycenters $\mathbf{S}_i = (\mathbf{S}, z_i^S)$ are calculated as for the ZEB-1 scheme above. Barycenter $\mathbf{S} = (\mathbf{S}, z^S)$ is added to the data set where z^S is found as a weighted sum of z-values z_i^S, but the weighting w_i is now according to the distance between $\mathbf{Q}_i$ and $\mathbf{S}_i$, i.e., in $3D$ space:

$$z^S = \sum_{i=1}^{3} \mathsf{w}_i \, z_i^S, \tag{8}$$

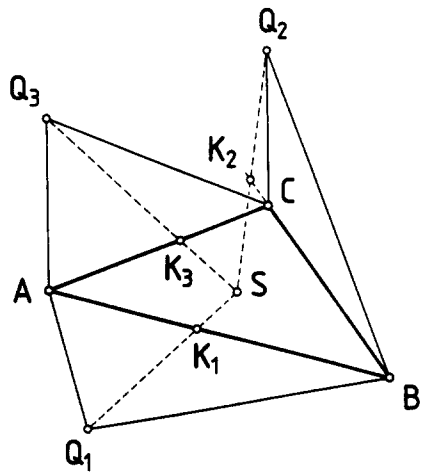


Figure 2. Notation of the two ZEB schemes.

where

$$w_i = \frac{1/(\|\mathbf{Q}_i - \mathbf{S}\|)}{\sum_{j=1}^{3}(1/\|\mathbf{Q}_j - \mathbf{S}\|)}.$$

- **Butterfly scheme.** This scheme is based on the interpolatory *butterfly subdivision scheme* of [14]. A point $\mathbf{S} = (\mathbf{S}, z^S)$ is added to the data set, found through the following procedure (cf. Figure 1).
  1. Calculate for edge a of triangle $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$ an auxiliary point $\mathbf{K}_a$ according to

$$\mathbf{K}_a = \frac{1}{2}(\mathbf{B} + \mathbf{C}) + 2\omega_a(\mathbf{A} + \mathbf{Q}_2) - \omega_a(\mathbf{Q}_1 + \mathbf{Q}_3 + \mathbf{R}_3 + \mathbf{R}_4), \tag{9}$$

     where design parameter ω_a is chosen such that $0 < \omega_a < 1/16$; smaller ω_a values create points $\mathbf{K}_a$ closer to edge a.
  2. Analogously for points $\mathbf{K}_b$ and $\mathbf{K}_c$ corresponding to triangle edges b and c.
  3. Define $\mathbf{S}$ as the average of points $\mathbf{K}_a$, $\mathbf{K}_b$, and $\mathbf{K}_c$:

$$\mathbf{S} = \frac{1}{3}(\mathbf{K}_a + \mathbf{K}_b + \mathbf{K}_c). \tag{10}$$

Note, since projections of points $\mathbf{K}_a$, $\mathbf{K}_b$, and $\mathbf{K}_c$ do not have to be inside of $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$, $\mathbf{S}$ might not be inside of $\Delta(\mathbf{A}, \mathbf{B}, \mathbf{C})$ too. In that case, we recalculate $\mathbf{S}$ for the new parameter values $\omega_a \mapsto \omega_a/2$, etc.

Note, if one of the triangle vertices is a convex hull vertex, one or even two of construction points $\mathbf{R}_j$ involved in the scheme might be missing eventually. In that case, they are defined (but only for the purpose of calculating point $\mathbf{S}$ and not to add them to the data set) via barycentric extrapolation. For example, if $\mathbf{A}$ is vertex of the convex hull, and $\mathbf{R}_6$ is missing, a *phantom point* $\hat{\mathbf{R}}_6$ can be defined through, e.g., orthogonal reflection or midpoint reflection of point $\mathbf{C}$ at edge $\mathbf{AQ}_3$. Our algorithm applies midpoint reflection which is very easy to realize using barycentric extrapolation with respect to triangle $\Delta(\mathbf{A}, \mathbf{C}, \mathbf{Q}_3)$, cf. Figure 1.

- **MQB-1 (Multiquadric Barycenter) scheme.** Barycenter $\mathbf{S} = (\mathbf{S}, z^S)$ is added to the data set where z^S is found through the multiquadric $MQ(\mathbf{x})$ (see, e.g., [4, Section 9.2.1]),

$$MQ(\mathbf{x}) = \sum_i \alpha_i \sqrt{d_i(\mathbf{x})^2 + R^2}, \qquad \text{with } d_i(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}\|, \tag{11}$$

  with constant R according to [15], which interpolates points $\mathbf{x}_i$ given by $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and the three vertices $\mathbf{Q}_i$ (cf. Figure 1) of the three direct neighbor triangles, i.e., $z^S = z^{mq} = MQ(\mathbf{S})$.

- **MQB-2 scheme.** Barycenter $\mathbf{S} = (\mathbf{S}, z^S)$ is added to the data set where z^S is found through the multiquadric $MQ(\mathbf{x})$, according to (11), which interpolates points **A**, **B**, **C**, the three vertices $\mathbf{Q}_i$ of the three direct neighbor triangles, and vertices $\mathbf{R}_i$ of their direct neighbors (cf. Figure 1). Note, since $\Delta(\mathbf{A},\mathbf{B},\mathbf{C})$ is an interior triangle, the existence of points $\mathbf{Q}_i$ can be guaranteed only (though two of them might coincide), but not all of the six additional points $\mathbf{R}_i$ might be available or distinct of each other in some situations. Therefore, the generation of the multiquadric of the MQB-2 scheme involves at the most twelve points.

Clearly, all these criteria and schemes are to a certain extend ad hoc and, we have to understand that it will be always possible to create data sets where the approximation over the data modified, augmented triangulation is even worse than the one over the original Delaunay triangulation.

2.3. Updating of the Delaunay Triangulation

To add a new point $\mathbf{P} = (\mathbf{P}, z^P)$, which has been calculated through one of the schemes to the data set and to actualize the Delaunay triangulation, first the new point is added to the end of the data list. Since **P** is per construction inside of $\Delta(\mathbf{A},\mathbf{B},\mathbf{C})$, we can split this triangle into the three triangles $\Delta(\mathbf{A},\mathbf{B},\mathbf{P})$, $\Delta(\mathbf{B},\mathbf{C},\mathbf{P})$, and $\Delta(\mathbf{C},\mathbf{A},\mathbf{P})$. In the triangle list $\Delta(\mathbf{A},\mathbf{B},\mathbf{C})$ is replaced by $\Delta(\mathbf{A},\mathbf{B},\mathbf{P})$ while $\Delta(\mathbf{B},\mathbf{C},\mathbf{P})$ and $\Delta(\mathbf{C},\mathbf{A},\mathbf{P})$ are added to the end of the list. Neighbors of all three triangles are determined and Lawson's LOP is applied to reassure the Delaunay triangulation. Here, according to [12], edges $\mathbf{A}P$, $\mathbf{B}P$, and $\mathbf{C}P$ are already *converged*, while edges $\mathbf{A}B$, $\mathbf{B}C$, and $\mathbf{C}A$ have to go through the LOP.

3. NUMERICAL TESTING

Testing was performed for 12 different test functions, $F_j(x,y)$, $j = 1(1)12$, defined on the unit square, exhibiting a wide variety of behavior. They were sampled over 24 different x-y data sets with sizes between 25 and 500 points. Most of the data sets and the first six of the test functions were kindly provided by R. Franke. The seventh test function was taken from [16] and two additional test functions were taken from [17]. These first nine test functions are also listed in [13]. The three last ones were created in analogy to functions given in [18]:

$$F_{10}(x,y) = \exp\left(-(1-(4x-2)(4y-2))^2\right), \tag{12}$$

$$F_{11}(x,y) = 1 - \exp\left(-(1-(4x-1)(4y-1))^2\right), \tag{13}$$

$$F_{12}(x,y) = \exp\left(-\left(4x+\frac{1}{4}\right)\left(4y^2-2y+1\right)+\left(y+\frac{1}{2}\right)\log\left(4x+\frac{1}{4}\right)\right). \tag{14}$$

Figures 3–5 display 3D-views and contour curve plots of test functions 1, 10, and 11.

$F_1(x,y)$ is Franke's famous exponential test function composed of two Gaussian peaks and a Gaussian dip, $F_{10}(x,y)$ is saddle shaped and represents a pass between two mountain ridges connecting two plateaus and, function $F_{11}(x,y)$ simulates a curved ravine forming the boundary of a planar area. Function $F_{12}(x,y)$ which is not illustrated is a gently rising hill which turns into a steep cliff.

For each of the $2D$ data sets the Delaunay triangulation T^D was created. Then augmented triangulations T^A were produced always using one of the detection criteria in combination with one of the point adding schemes. Approximation quality of $f_{T^A,F}$ was compared with the one of $f_{T^D,F}$ visually and numerically: first, $3D$ plots and contour plots of $f_{T^A,F}$ were generated and compared with $3D$ plots and contour plots of $f_{T^D,F}$ as well as with corresponding plots of the test functions. Second, errors between the PLIs $f_{T^D,F}$ and $f_{T^A,F}$ and the test functions were computed on a grid of 50×50 nodes uniformly placed over the unit square. Figures 6–8 show for test functions of Figures 3–5 and various data sets the initial Delaunay triangulation T^D, (a), the data modified augmented Delaunay triangulation T^A, (b), $3D$ perspective views of $f_{T^D,F}$, (c),

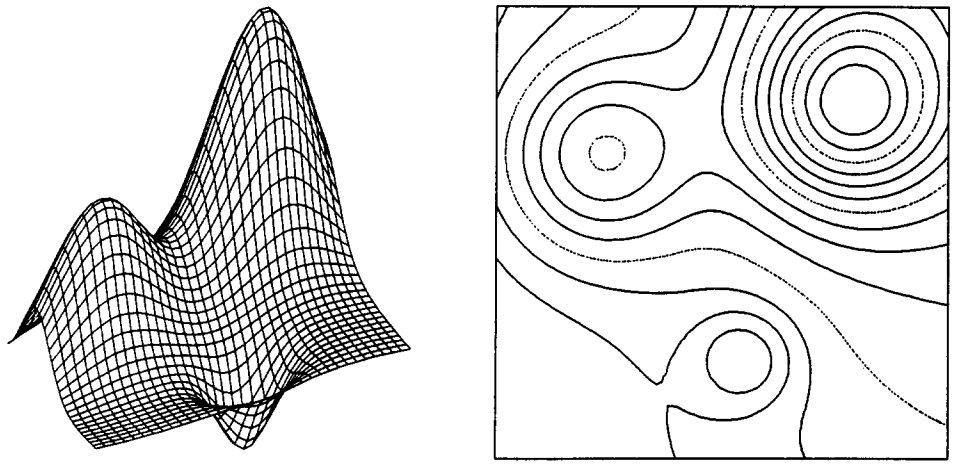


Figure 3. Perspective view and contour curves of test function $F_1(x, y)$.

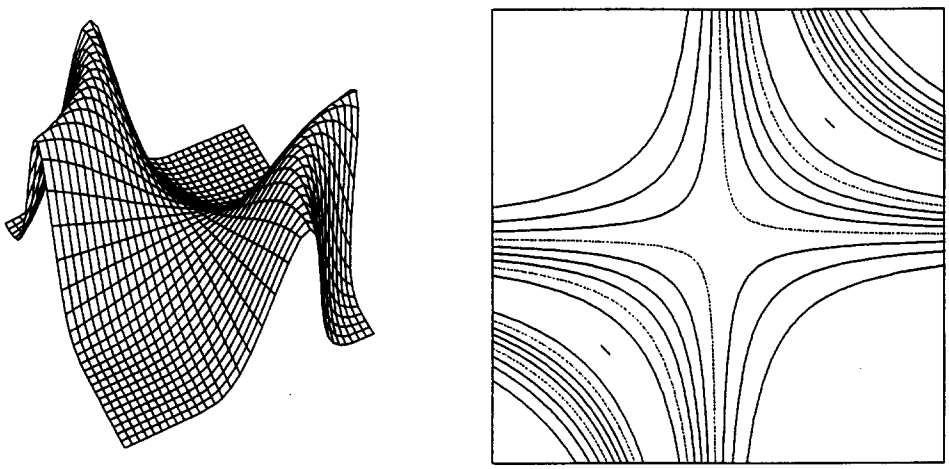


Figure 4. Perspective view and contour curves of test function $F_{10}(x, y)$.

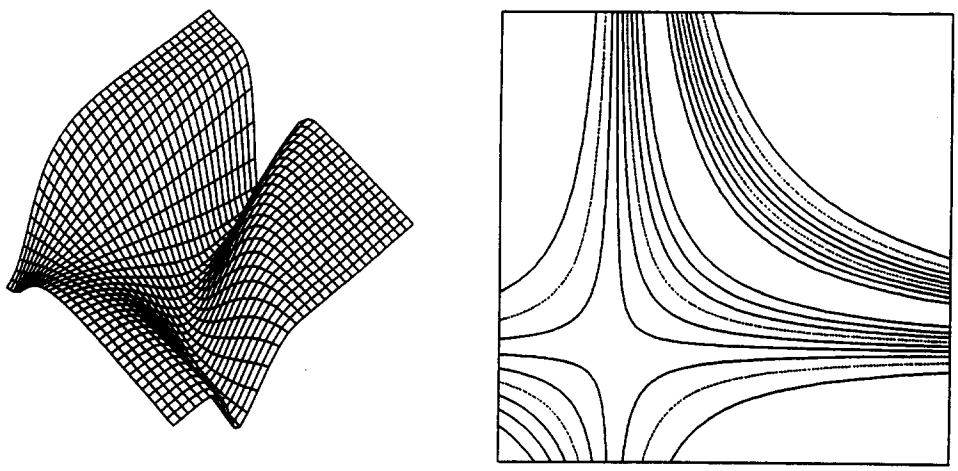
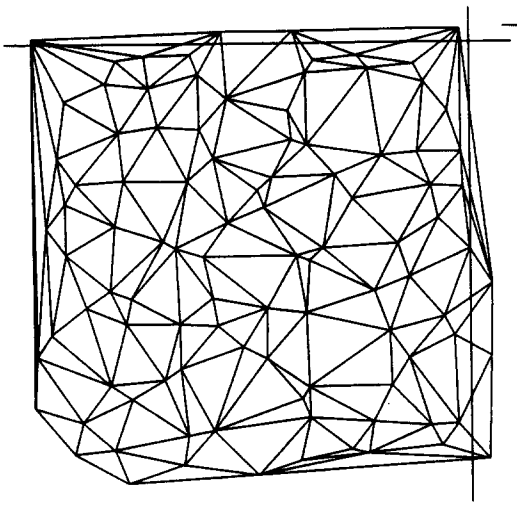
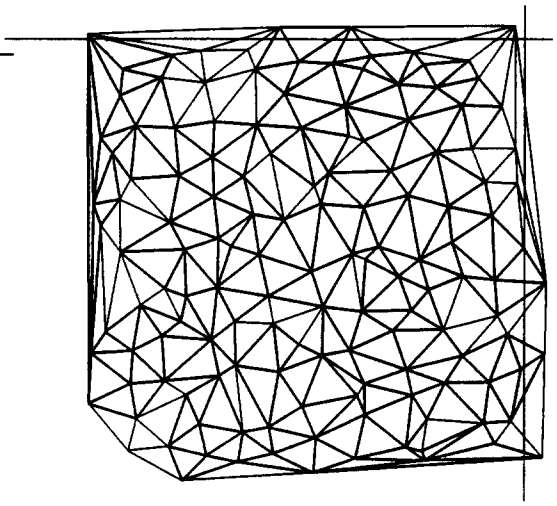


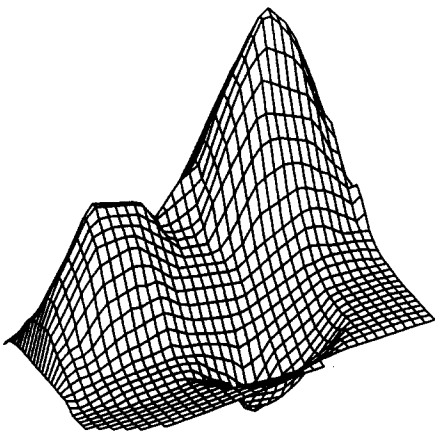
Figure 5. Perspective view and contour curves of test function $F_{11}(x, y)$.



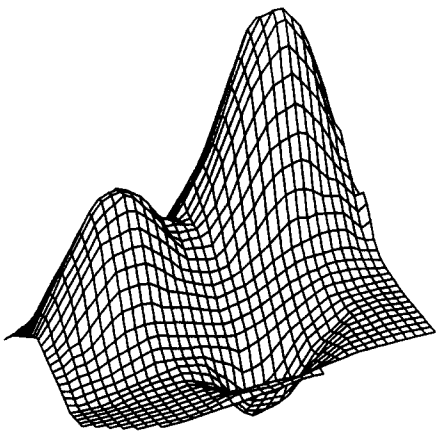
(a) Delaunay triangulation T^D.



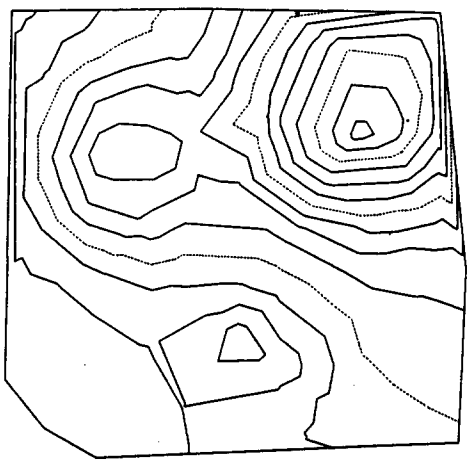
(b) Augmented Delaunay triangulation T^A.



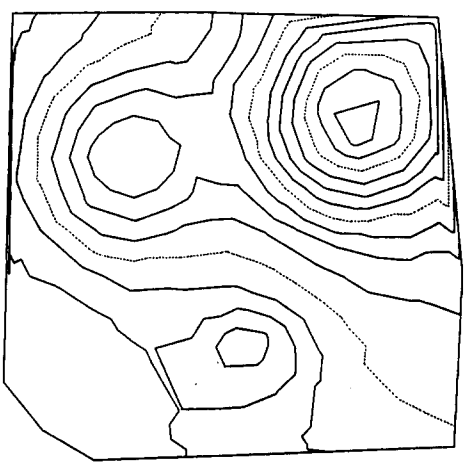
(c) PLI $f_{T^D,F}$.



(d) PLI $f_{T^A,F}$.

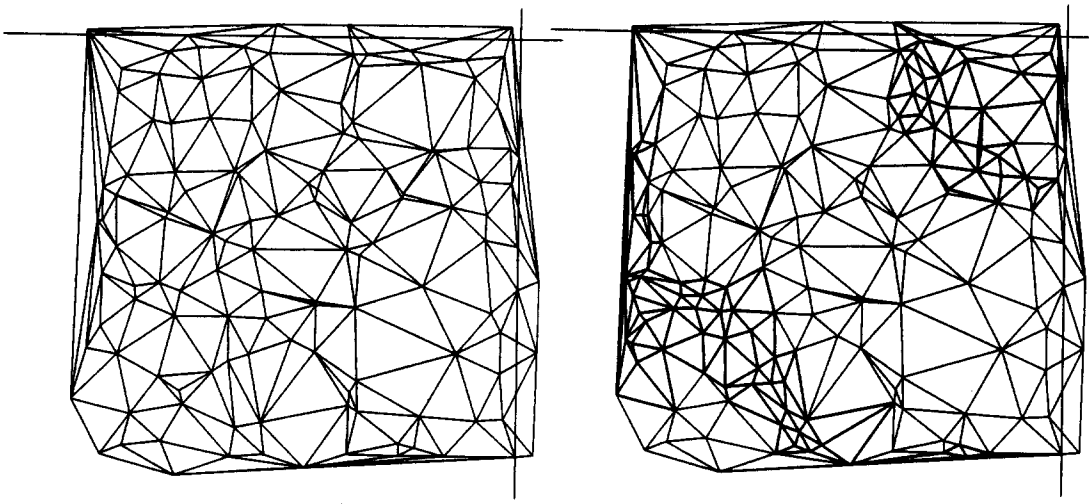


(e) Contour curves of $f_{T^D,F}$.



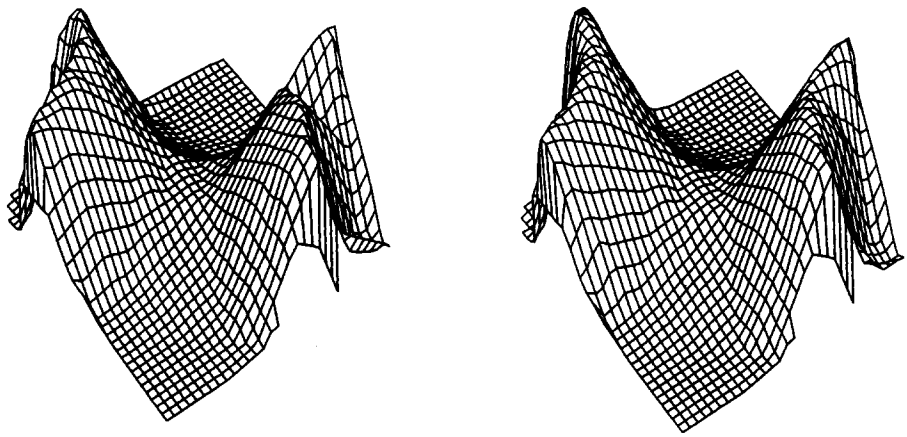
(f) Contour curves of $f_{T^A,F}$.

Figure 6. Delaunay triangulations T^D and T^A, PLIs $f_{T^D,F}$, and $f_{T^A,F}$ and corresponding contour plots based on 100 data points (Franke's ds3) and the SA/MQB-2 strategy using $T_A = 1.3$. Max/Mean/RMS error reduction is 0/40/41 % with 40 points added to the data set.



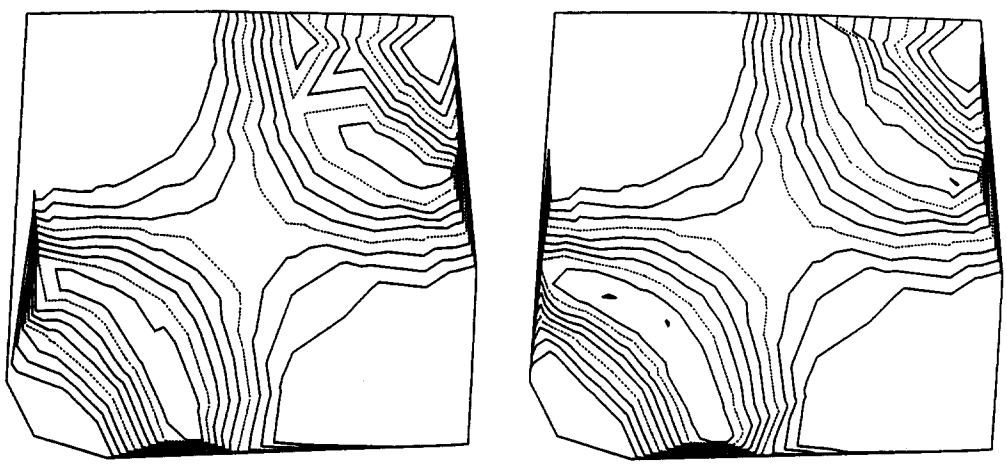
(a) Delaunay triangulation T^D.

(b) Augmented Delaunay triangulation T^A.



(c) PLI $f_{T^D,F}$.

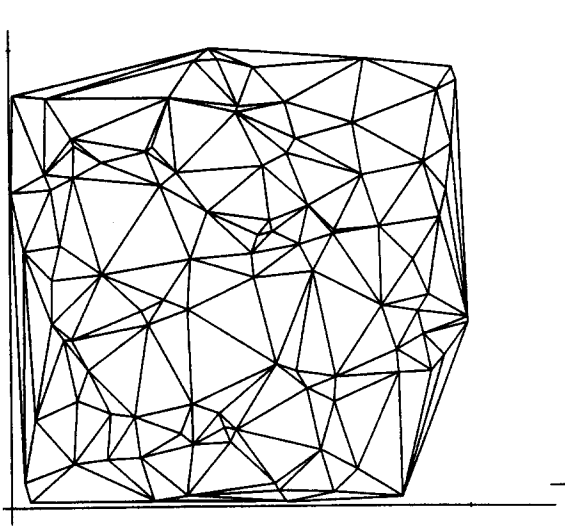
(d) PLI $f_{T^A,F}$.



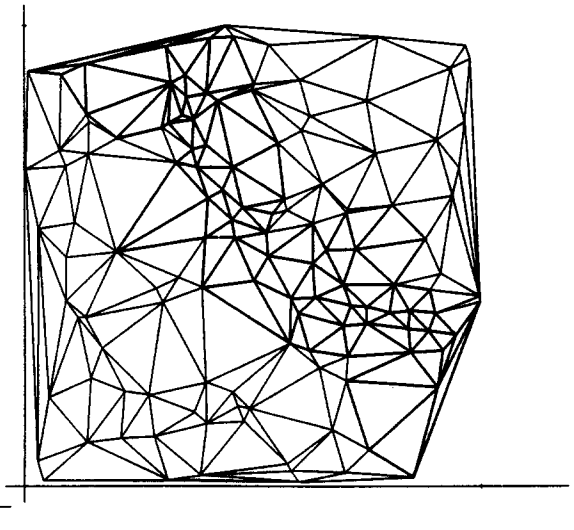
(e) Contour curves of $f_{T^D,F}$.

(f) Contour curves of $f_{T^A,F}$.

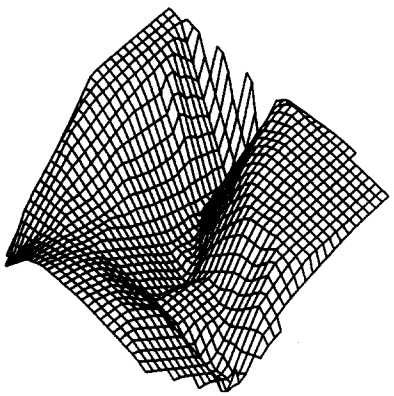
Figure 7. Delaunay triangulations T^D and T^A, PLIs $f_{T^D,F}$, and $f_{T^A,F}$ and corresponding contour plots based on 130 data points (Franke's fault7) and the STA/MQB-2 strategy. Max/Mean/RMS error reduction is 7/29/35% with 43 points added to the data set.



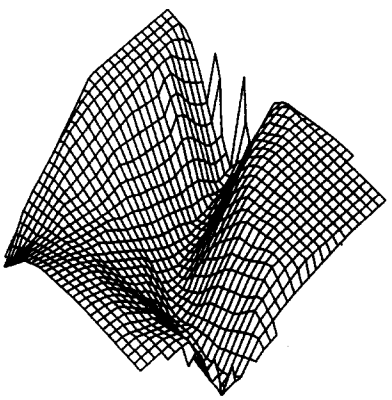
(a) Delaunay triangulation T^D.



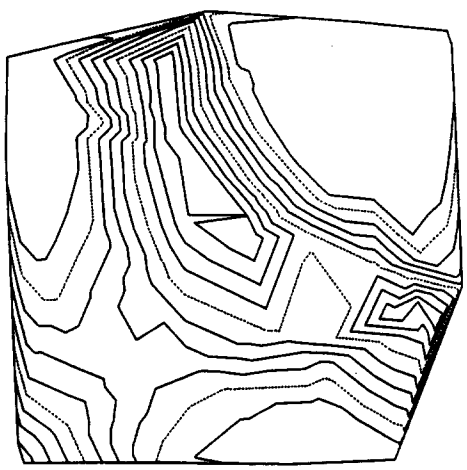
(b) Augmented Delaunay triangulation T^A.



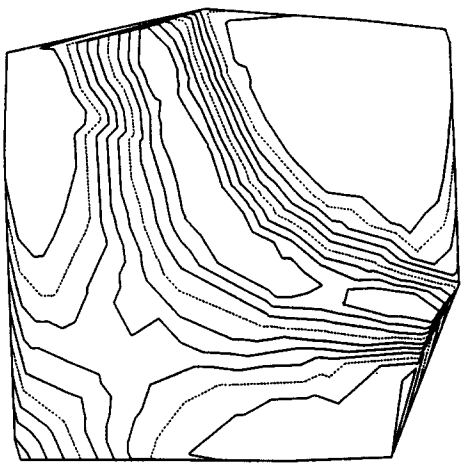
(c) PLI $f_{T^D,F}$.



(d) PLI $f_{T^A,F}$.



(e) Contour curves of $f_{T^D,F}$.



(f) Contour curves of $f_{T^A,F}$.

Figure 8. Delaunay triangulations T^D and T^A, PLIs $f_{T^D,F}$, and $f_{T^A,F}$ and corresponding contour plots based on 100 data points (Franke's ds100) and the STA/MQB-2 strategy. Max/Mean/RMS error reduction is 5/37/34% with 30 points added to the data set.

and of $f_{T^A,F}$, (d), and the contour curves of $f_{T^D,F}$, (e), and of $f_{T^A,F}$, (f). Changes in the triangulation are highlighted in bold. Please note, changes and improvements can be expected mainly in the interior area of the convex hull!

The first three modification schemes turned out to be not very suitable for the purpose of improving the approximation property of the PLI, with any of the three triangle detection criteria. The approximation of the PLI of the data modified augmented Delaunay triangulation, $f_{T^A,F}$, was often not as good as the one of the initial Delaunay triangulation, i.e., of $f_{T^D,F}$. Especially, the butterfly scheme seems to be inappropriate. Behaviour of ZEB-1 and ZEB-2 is somehow predictable at least. Deteriorations appear always in case that long and thin neighbor triangles are present and/or the test function demonstrated strong curvature changes in the local neighborhood.

The two MQB schemes performed much better. They are capable of removing badly shaped interior triangles of the Delaunay triangulation and they have the capability of improving the approximation of test functions. Though, MQB-1 did not perform very well in combination with the area criterion. It looks like that long extrapolation distances are the reason for this result as well as a shortage of data to calculate local multiquadrics to be relied upon. MQB-1 performed pretty good in combination with the TA criterion. Improvement of (Max, Mean, RMS) error values was up to (63%, 74%, 73%) and improvement was achieved in (57%, 80%, 89%) of all tests. Remarkably high are also the numbers of average percentages of improvements, (19%, 14%, 15%) of (Max, Mean, RMS) errors. These are the changes to be expected in general. Although there were 25 cases were the mean error increased, 18 of these cases happened with test function $F_7(x, y)$ which represents a sharp peak on a plan and a rather steep rising ramp leading to another plane. Obviously, the angle threshold of 35° is too small for this test function.

Not all 15 possible combinations of detection criteria and modification schemes were tested on all 288 combinations of test functions and data sets—some quickly proved not to perform very well—but altogether we were running about 2000 different tests with threshold values $T_A = 3$ and $T_\phi = 35°$. Tables 1–3 listed below summarize the tests with three of the more successful strategies: STA/MQB-1, SA/MQB-2, and STA/MQB-2. The tables display the number of tests performed and, for the maximum (Max), mean (Mean), and Root Mean Square (RMS) errors, the absolute numbers of tests with improvements and with deteriorations over the $f_{T^D,F}$ approximation. Furthermore are listed for Max, Mean, and RMS errors: best and worst changes relative to the PLI $f_{T^D,F}$, the average percentage of improvement and of deterioration, as well as the overall average percentage of change. Negative values indicate improvement, i.e., reduction of the corresponding error, positive values indicate deterioration, i.e., increase of the error. Note, the total number of improvements and of deteriorations do not always sum to the total number of tests since in some cases no changes of the specific error measurement took place for the chosen threshold values and combination of detection criterion and modification scheme.

Table 1. Statistic of the STA/MQB-1 strategy.

Error	Tests #	Improvements			Deteriorations			Overall Average %
		#	Max %	Average %	#	Max %	Average %	
Max	168	101	−62.9	−23.4	8	55.3	16.5	−13.3
Mean		156	−67.2	−14.5	12	26.7	9.3	−12.8
RMS		153	−64.4	−18.3	15	17.5	5.2	−16.2

Output for the MQB-1 scheme using the STA criterion is listed in Table 1. This combination performed even better than the TA/MQB-1 strategy. Now improvement of (Max, Mean, RMS) errors was achieved in (60%, 93%, 91%) of all tests! Average percentage of improvement was also higher for all three error measurements and there were less deteriorations and the percentages of worst and of average deteriorations are both smaller. Deteriorations, when present, could be traced back to situations were long and thin boundary triangles were in the direct neighborhood of the triangles under consideration and, generally spoken, for small data sets, i.e., sets with

Table 2. Statistic of the SA/MQB-2 strategy using threshold $T_A = 3$.

Error	Tests #	Improvements			Deteriorations			Overall Average %
		#	Max %	Average %	#	Max %	Average %	
Max	168	57	−49.1	−16.9	5	50.3	17.6	−5.2
Mean		121	−73.5	−20.4	11	29.1	12.7	−13.9
RMS		125	−63.8	−19.0	8	20.2	7.8	−13.8

Table 3. Statistic of the STA/MQB-2 strategy.

Error	Tests #	Improvements			Deteriorations			Overall Average %
		#	Max %	Average %	#	Max %	Average %	
Max	168	104	−67.1	−25.3	5	50.2	14.6	−15.2
Mean		161	−73.5	−17.8	7	21.9	6.7	−16.8
RMS		161	−67.3	−21.1	7	16.2	3.5	−20.1

less than 50–60 points, so that there is a lack of data information to build up reliable local multiquadrics.

Performance of MQB-2 in combination with the surface area criterion was somehow between TA/MQB-1 and STA/MQB-1. Table 2 summarizes the results. Here again $F_7(x, y)$ was the *killjoy*. Eight out of 11 cases of increasement of the mean error occurred for this test function.

In combination with the TA criterion results were much better for MQB-2. They even topped the outcome of the STA/MQB-1 strategy documented by Table 1. Only test function $F_7(x, y)$ was a *trouble-maker* again: mean error increased all together in 15 cases out of 168, but 12 of these cases occurred with $F_7(x, y)$.

Best performance was given by MQB-2 in combination with STA (see Table 3). Now, improvement of (Max, Mean, RMS) errors was attained in (62%, 96%, 96%) of all tests. In this context, it is very remarkable that STA/MQB-2 usually adds fewer points to a data set. TA/MQB-2 is the only strategy which adds fewer points than STA/MQB-2 (using $T_\phi = 35°$).

4. SUMMARY AND FUTURE WORK

We introduced and tested several criteria to detect badly shaped interior triangles and schemes to add interior data to the original data set. Our aim was to enhance the interior approximation quality of a PLI defined with respect to the Delaunay triangulation of the data set. Not all of the tested detection criteria and point schemes fulfilled the expectations but some did very well and will be subject of continuous work.

The two multiquadric schemes performed best and out of the six possible combinations of multiquadric schemes and detection criteria only the strategy SA/MQB-1 cannot be recommended. STA/MQB-2 is the best performer followed by strategy TA/MQB-2, STA/MQB-1 is coming in third. It is very interesting that strategy STA/MQB-2 not only is the most progressive strategy but it also forms the most conservative one. Progressive in the sense that on an average it results in most error reductions and at the same time results in the biggest error reductions of the PLI approximation. Conservative in the sense that on an average it results in fewest error increments and also in smallest increments of all error measurements, and it usually adds fewer points to a data set than other strategies do (only TA/MQB-2 adds fewer points).

It is important to note that, since all our strategies change interior triangles, only improvement in the interior area can be expected! At first hand the boundaries of the triangulation and of the PLI are not changed. Although, occasionally, boundary triangles are changed too through the LOP (see Section 2.3). However, primarily there will be no major improvement of the approximation property of the PLI in the boundary area. Nevertheless, peaks of the Max error

are often situated exactly in the boundary area and the boundary contribution to Mean and RMS errors is high too, in the presence of long and thin boundary triangles, which appear quite often. Most of the time long and thin boundary triangles actually have bad influence on the interior improvement too. We will consider this important point later again. For right now we are able to conclude that our results are excellent under these circumstances.

We have to realize and understand: due to the nature of the scattered data problem there can be no strategy which yields improvement for any kind of data set! Though, the recommended strategies performed quite well with only a few data configurations where deteriorations occurred. Since we found out that these configurations were somehow predictable, we got rather fast a very good sense of how to work successfully with the software. Yet, there are possibilities for further and future development of the introduced and described ideas.

First, to reduce the percentage of cases in which deterioration, really the opposite of what we were aiming for, was the result, together with J. Braun we are presently investigating the possibility of an *'Undo function'*. The idea is to perform changes of the data set only in situations in which improvement is achieved. While this plan is easily carried out and works fine if we are dealing with test functions, real scattered data originating from measurements create more serious headaches.

Second, since there is a strong dependence of the amount of improvement on the threshold, the question is how to choose the optimal one—optimal in the sense that the error reduction becomes as big as possible with the smallest number of extra points added to the data set. First investigations indicate that typically the scenario is the following: for the SA/MQB-2 strategy for example the error reduction function qualitatively shows the behaviour of Figure 9, i.e., for decreasing values of the threshold variable T_A an increasing number of points is added to the data set, also increasing the error reduction. But compared with the boundary correction described in [11], more points have to be added to the data set to get the same error reduction percentage. We also noticed that it is more difficult to achieve improvement of the Max error measurement.

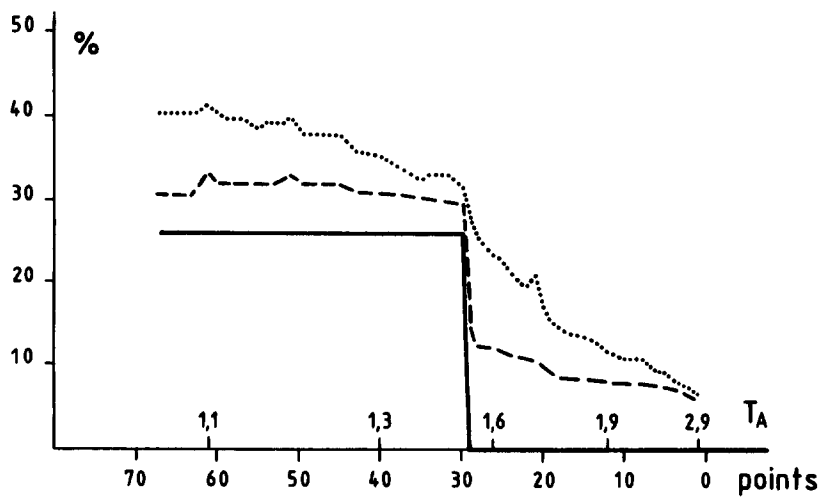


Figure 9. Max (bold), Mean (dotted), and RMS (dashed) error reduction for test function $F_3(x, y)$, Franke's 100 point data set ds3 and strategy SA/MQB-2.

Third, it might be very advantageous and improve the approximation quality a lot and with fewer points added, to apply various triangle detection criteria, various modification schemes, and various thresholds locally. That means for different areas—may be even for different triangles—of one and the same data set. This is obvious, for example, in context of rapidly varying functions such as test functions $F_{10}(x, y)$ and $F_{11}(x, y)$.

Fourth, in view of remarks two and three various procedures of how to successively handle the list of pinpointed badly shaped interior triangles might be advantageous. In the moment we

simply deal with them in the order they show up through the detection and because of the initial triangle list implied by the data base used in Lawson's algorithm. Of course there might be better ways to do it. Jointly with Braun we are currently looking into several different ideas of sorting and successively proceeding interior triangles. First tests indicate already major performance gains.

Fifth, to improve usability of the program, we need automation with respect to choosing detection criteria, modification scheme and optimal threshold to result in the best possible PLI. This is especially necessary if we want to introduce localization according to point two. Automation via the number of points added or the error reduction instead of the threshold might be an alternative procedure. The STA criterion is already a first step towards automation. Because of its success we are going to implement a SSA (Stochastic Surface Area) criterion similarly to the STA criterion.

Sixth, as pointed out earlier, the modified Delaunay triangulation might still contain badly shaped boundary triangles giving reason for a poor boundary approximation property. Therefore, quality might be improved further by modification of boundary triangles. To detect and correct badly shaped boundary triangles, we already developed three detection criteria and seven modification schemes. Some of them resulted in as impressive improvements as the criteria and schemes described above and are documented in [10,11] (boundary correction for PLI defined over data dependent triangulations has been studied first in [19]). We actually observed that long and thin boundary triangles can have a very bad influence on the interior correction and, that interior correction might *'reshape'* long and thin boundary triangles into even thinner boundary and near-boundary triangles. Because of this the Max error measurement often is not improved, not improved a lot, or at least not until relatively many points have been added to the data set (cf. examples of Figures 6–8 and Figure 9).

Seventh, applying several different interior modifications in a row might turn out to be advantageous. Boundary and interior modifications might also be combined, multiplying the power of each of the two methods. And indeed, some early tests clearly showed that the negative influence of badly shaped boundary triangles on the interior improvement, as described above, can be eliminated by employing boundary correction first followed by interior correction. Furthermore, many of the strategies for interior improvement gave much better results after long and thin boundary triangles had been removed initially through a boundary correction preprocessing step. Results are described in [20].

Eighth, the methods described might be applied on piecewise linear, quadratic, etc. interpolants with respect to a data dependent triangulation. While this is mainly subject of ongoing work, first results are already documented in [21].

Ninth, originally starting with a C-code *test program*, for several reasons now all routines have been successfully implemented in C++ such that they easily carry over to the higher dimensional situation. Since many of the detection criteria and point adding schemes can be generalized too, this will be one of our next steps to do.

REFERENCES

1. L.L. Schumaker, Fitting surfaces to scattered data, In *Approximation Theory II*, (Edited by G.G. Lorentz, C.K. Chui and L.L. Schumaker), pp. 203–268, Academic Press, New York, (1976).
2. R. Franke, Scattered data interpolation: Tests of some methods, *Mathematics of Computation* **38** (157), 181–200, (1982).
3. R. Franke and G.M. Nielson, Scattered data interpolation and applications: A tutorial and survey, In *Geometric Modeling*, (Edited by H. Hagen and D. Roller), pp. 131–160, Springer, Berlin, Heidelberg, New York, (1991).
4. J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, AK Peters, Wellesly, (1993).
5. R.J. Renka and A.K. Cline, A triangle-based C^1 interpolation method, *Rocky Mountain Journal of Mathematics* **14** (1), 223–237, (1984).
6. A. Preusser, Algorithm 684. C^1 and C^2 interpolation on triangles with quintic and nonic bivariate polynomials, *ACM Transactions on Mathematical Software* **16** (3), 253–257, (1990).

7. J. Bramble and M. Zlamal, Triangular elements in the finite element method, *Mathematics of Computation* **24**, 809–820, (1970).
8. J.A. Gregory, Error bounds for linear interpolation in triangles, In *The Mathematics of Finite Elements and Applications II*, (Edited by H.J.H. Whitemann), pp. 163–170, Academic Press, London, (1975).
9. R.E. Barnhill and J.A. Gregory, Polynomial interpolation to boundary data on triangles, *Mathematics of Computation* **29** (131), 726–735, (1975).
10. T. Stüttgen, Datenoptimierte Delaunay-Triangulierungen, Diplomarbeit, Universität-Gesamthochschule-Duisburg, (1995).
11. D. Lasser and T. Stüttgen, Boundary improvement of piecewise linear interpolants defined over Delaunay triangulations, *Computers Math. Applic.* **32** (10), 43–58, (1996).
12. C.L. Lawson, Software for C^1 surface interpolation, In *Mathematical Software III*, (Edited by J.R. Rice), pp. 161–194, Academic Press, Boston, (1977).
13. N. Dyn, D. Levin and S. Rippa, Data dependent triangulations for piecewise linear interpolation, *IMA Journal on Numerical Analysis* **10**, 137–154, (1990).
14. N. Dyn, D. Levin and J.A. Gregory, A Butterfly subdivision scheme for surface interpolation with tension control, *ACM Transactions on Graphics* **9** (2), 160–169, (1990).
15. S.E. Stead, Estimation of gradients from scattered data, *Rocky Mountain Journal of Mathematics* **14** (1), 265–279, (1984).
16. S.I.M. Ritchie, Surface representation by finite elements, Ms. Thesis, University of Calgary, (1978).
17. T. Lyche and K. Morken, Knot removal for parametric B-spline curves and surfaces, *Computer Aided Geometric Design* **4** (3), 217–230, (1987).
18. E. Castillo and A. Iglesias, Some applications of functional equations to the characterization of families of surfaces, In *Proceedings of the first Peruvian Workshop on CAGD, CAGD '94*, (Edited by D. Lasser), pp. 153–169, Shaker, Aachen, (1995).
19. N. Dyn, D. Levin and S. Rippa, Boundary correction for piecewise linear interpolation defined over data dependent triangulation, Report of the School of Mathematical Sciences, Sackler Faculty of Exact Sciences, Tel-Aviv University, (1991).
20. D. Lasser, T. Stüttgen and J. Braun, Piecewise linear interpolation over data modified Delaunay triangulations, (in preparation)(1996).
21. S. Kropac, Datenoptimierte Datenabhängige Triangulierungen, Diplomarbeit, Universität-Gesamthochschule-Duisburg, (1995).